



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/779,298	02/13/2004	Svetlana Loksh	MS306137.01 / MSFTP550US	7422
27195 7590 06/25/2008 AMIN. TUROCY & CALVIN, LLP 24TH FLOOR, NATIONAL CITY CENTER 1900 EAST NINTH STREET CLEVELAND, OH 44114			EXAMINER HUQ, AHMED E	
			ART UNIT 2192	PAPER NUMBER
			NOTIFICATION DATE 06/25/2008	DELIVERY MODE ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket1@thepatentattorneys.com
hholmes@thepatentattorneys.com
lpasterchek@thepatentattorneys.com

Office Action Summary	Application No. 10/779,298	Applicant(s) LOKSH ET AL.	
	Examiner AHMED E. HUQ	Art Unit 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 26 March 2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-9 and 11-34 is/are pending in the application.
- 4a) Of the above claim(s) 10 is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-9 and 11-34 is/are rejected.
- 7) ☒ Claim(s) 4 is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Remarks

1. Applicants Amendment and response dated March 26, 2008, responding to the Dec 26, 2007 office action provided in the rejection of claims 1-23, where claims 1, 6, 15, 24, 29 and 32 has been amended. Claim 10 has been cancelled. Claims 1-9, and 11-34 remain pending in the application and which have been fully considered by the examiner.

Applicant arguing for the claims are patentable over Herrmann (*see page 11-14 of the amendment and response*) primarily based on assertions in particular at page 12, that Herrmann neither discloses nor suggests applicants' Claim 1, as amended to include with “ *a host adaptor that merges at least one menu of the unmanaged application with at least one menu of the development environment, **determines if the development environment has focus or if unmanaged application has focus, and enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination***” are not persuasive, as will be addressed under Prior Art's arguments- Rejections section at item 2 below.

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Prior Art's Arguments – Rejections

2. Claims 1-9 and 11-34 are currently pending. Claims 1,6, 15,24,29, and 32 have been amended. Claim 10 has been cancelled. Objection of drawing has been considered by examiner. Applicant has amended claim 1 to over come the 101 rejection has been considered.

Applicant's arguments files on March 26, 2008, in particular on pages 11-14 have been fully considered, but they are not persuasive. See below for examples:

At page 12, Applicant contend that Claim 1 is not anticipated by Herrmann, as Herrmann does not disclose that the “ *A system that facilitates a software development environment, comprising: a hosting component that interfaces an unmanaged application... and a **host adaptor that** merges at least one menu of the unmanaged application with at least one menu of the development environment, **determines if the development environment has focus or if unmanaged application has focus, and enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination**” (see page 12, of the amendment and*

response) as the instance application invention has recited and/or indicated in claim 1, which examiner strongly disagree. A closer reading of Herrmanns' art reveals, indeed, disclose a method covering the step of "*A system that facilitates a software development environment, comprising: a hosting component that interfaces an unmanaged application* (Fig. 3, form 371, Col. 3, lines 21-30 where ActiveX and Delphi are such example of an unmanaged application)... *and a **host adaptor*** (Fig. 3, main window 361) ***that merges at least one menu of the unmanaged application with at least one menu of the development environment, **determines if the development environment has focus***** (Fig. 3, view of a focus screen such as: form 371 on a environment which displays the form 371 is active and other applications are inactive) ***or if unmanaged application has focus*** (Col. 6, lines 43-47), ***and enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination*** "(E.g. see Fig. 3 illustrates an application development environment 360, comprises a main window 361, a form 371, a code editor window 381, and an object manager window 391, col. 6 lines 40-58, which states "...the main window 361 (host adaptor). The form can be thought as a component that contains other components (such as: toolbar). One form serves as the main form for the application; its components interact with other forms and their components to create the interface for an application under development... while other forms typically serves as dialog boxes, data entry screens, and the likes..." (emphasis added)

In response to applicant's argument in Claim 1, at page 13 that the references fail to show certain features of applicant's invention, it is noted that the features upon

which applicant relies (i.e., “ Moreover, determinations on items to enable/disable and implementation of the determination is done automatically..”) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Applicant also argues Herrmann and Dasilva does not disclose a system that facilitates an IDE in claim 15, as ***“enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination”*** (page 13, section IV.) Examiner points out that Harrmann, indeed, discloses such claimed limitations (E.g. see Fig. 3 illustrates an application development environment 360, comprises a main window 361, a form 371, a code editor window 381, and an object manager window 39; Col. 6 lines 40-58, which states “...the main window 361 (host adaptor). The form can be thought as a component that contains other components (such as: toolbar). One form serves as the main form for the application and its components interact with other forms and their components to create the interface for an application under development which typically serves as dialog boxes, data entry screens”; Col. 6, lines 43-55 where forms 371 are the focal point of nearly every application which one develops in the environment; Col. 6, lines 59 to col. 7, lines 26, when the system in ‘design’ mode user can change the properties of the form, features, properties, by using object inspector window 391 to edit the form during design time.) (emphasis added).

In response to applicant's argument in Claim 15, at page 14 that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., " automatically enabling and disabling different menu items...") are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Claim Objections

Claim 13 objected to because of the following reason: According to Amended Claim 1, a system wherein memory coupled to a processor.., but Claim 13, it states as 'computer' . Appropriate correction is required.

Claims 33 and 34, should states a 'system' claim NOT 'method' claim.
Appropriate correction is required.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 15-23 and 32-34 rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 15, recites a "system" that has been reasonably interpreted as computer software, program, component, listings per se. Claim 15 fails to recite the "system" as stored on an appropriate computer readable medium, which defines structural and functional interrelationships between the software and other components of a computer

that permit the software's functionality to be realized-see MPEP 2106.01(I), Therefore, Claim 15 is rejected as non-statutory.

Claims 16-23 mirror the deficiencies of claim 15 and are also rejected as non-statutory.

Claim 32, recites a "system" that has been reasonably interpreted as computer software means, program, component, listings per se. Claim 32 fails to recite the "system" as stored on an appropriate computer readable medium, which defines structural and functional interrelationships between the software and other components of a computer that permit the software's functionality to be realized-see MPEP 2106.01(I), Therefore, Claim 6 is rejected as non-statutory.

Claims 33 and 34 mirror the deficiencies of claim 32 and are also rejected as non-statutory.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-9 and 11-34 are rejected under 35 U.S.C. 102(b) as being anticipated by Herrmann (US Pat No. 5,995,756)

Regarding claim 1, Herrmann discloses a system that facilitates a software development environment, comprising:

~~an unmanaged application that facilitates processing a document; and a hosting~~

component (Fig. 3, main window 361) that interfaces an unmanaged application (Fig. 3, form 371, Col. 3, lines 21-30 where ActiveX and Delphi are such example of an unmanaged application); a designer framework that interfaces a development environment (Fig. 3, Programming environment 360), the hosting component and designer framework interface such that the unmanaged application functions (Col. 6, lines 29-42 where application functions lists are located in tool bar buttons 363, main menu 362 and) as a designer within the development environment (Fig. 3, , component palette 364); and a host adaptor that merges at least one menu of the unmanaged application (in view of a focus screen, form 371 on a environment its displays the form 371 active) with at least one menu of the development environment (Col. 6, lines 33-42 where main menu 362 lists user-selectable command...), determines if the development environment has focus Or if unmanaged application has focus (col. 6, lines 43-47), and enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination (col. 6, lines 29-42, 50-58); wherein memory coupled to a processor retains the hosting component, the design framework, the host adaptor, or a combination thereof. (Col. 4, lines 60-67 where a main memory 102, a mass storage 107 in FIG 1B).

Regarding to claim 2, Herrmann teaches *hosting component supports an object linking and embedding (OLE) technology* (col. 2, lines 39-44, Col. 10, lines 1-15 where an ActiveX Control embedded within an HTML page)

Regarding to claim 3, Herrmann teaches *the unmanaged application includes at least one of a word processing application [[and]] or a spreadsheet application* (Col. 2, lines 55-65 where an active document such as: Microsoft Word Document requires that a host application 'MS word' be present locally on the user's machine).

Regarding to claim 4, Herrmann teaches *an integration interface to facilitate integrating a third-party unmanaged application as a designer in the development environment* (col. 7, lines 1-14 where third party as dialog boxes in a form of an application.)

Regarding to claim 5, Herrmann teaches *the development environment is an integrated development environment* (FIG 2, 211. col. 5, lines 36-45).

Regarding to claim 6, Herrmann teaches ~~further comprising a host adaptor that interfaces to the unmanaged application, which~~ the host adaptor is application specific to facilitate integrating the unmanaged application functioning as a designer within ~~into~~ the development environment (see FIG 1B, development computer software system 150, col. 5, lines 7-25, 38-46, where system 200 includes components which interface with the system 100 through windows shell 180 and OS 160, moreover, host adaptor such as: compiler; Fig. 3, Delphi which includes commands and IDE.)

Regarding to claim 7, Herrmann teaches *a document-hosting subcomponent that facilitates hosting the document that can be manipulated in the development environment* (Col. 6, lines 29-44 where main menu 362, tool bar buttons 363, component palette 364 are subcomponent of an application Fig 3, 360).

Regarding to claim 8, Herrmann teaches *the document can be manipulated using at least one of the unmanaged application and the development environment* (Col. 2 lines 55-65 where Microsoft Word Document).

Regarding to claim 9, Herrmann teaches *the document is one of a new document and an existing document* (col. 3, lines 1-20 where an existing document is distributed application through web page and new document is an active document that can be view on-screen representation.).

Regarding to claim 11, Herrmann teaches *facilitates hosting a plurality of different unmanaged applications* (col. 2 line 60-65, col 3. line 21-35, col. 5 lines 38-50 where as plurality of different unmanaged applications HTML, ActiveX, MIME, compiler....)

Regarding to claim 12, Herrmann teaches *a computer readable medium having stored thereon computer executable instructions for carrying out the system* (col. 1. col 2. lines 1-65 where the computer readable storage is HTTP server and carry out instruction is TCP/IP link, Fig. 1A system 100, mass storage 107....)

Regarding to claim 13, Herrmann teaches *a computer employing the system* (See Fig 1A computer system 100, 1B computer software system 150, Col. 5, lines 6-15, where it teaches all the components in the system is part of the computer. Examiner treats computer as a whole.)

Regarding to claim 14, Herrmann teaches *a tray component that facilitates presenting non-visual data in the IDE* (Col. 6, lines 59-67 where in design mode of the

system, the user can change the properties of the form..., Col. 7 lines 29-35 where as tray components are toolbar, main menu; user can navigate within the application.)

Regarding to claim 15, Herrmann teaches ~~an unmanaged application that facilitates processing a document; and a hosting component that interfaces to the unmanaged application such that the unmanaged application functions as a designer within the IDE~~ (Col. 6, 30-47 where hosting adaptor is consist of all the component in Fig. 3 an application development environment 360), *the hosting component further comprising,*

an integration interface to facilitate integrating a third-party unmanaged application as a designer in the development environment (Col. 7, lines 9-26 where Microsoft Windows common dialog boxes can be installed as third-party component);

a host adaptor that interfaces to the unmanaged application, ~~which~~ the host adaptor is application specific to facilitate integrating the unmanaged application into the development environment, merges at least one menu of the unmanaged application with at least one menu of the development environment (Col. 6, lines 33-42 where main menu 362 lists user-selectable command...), *determines if the development environment has focus **or** if unmanaged application has focus, and enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination;* and (col. 6, lines 29-42, 51-58);

a document-hosting subcomponent that facilitates hosting a document that can be manipulated in the development environment (col. 8, lines 10-25 where the host adaptor is MIME and associate components such as: OLE, GUID Class ID.), *the hosted*

document has at least two views, including a design view and a code view; and[[.]] (Fig. 3, inspector window 391 'design view' and code editor 381 'code view')
a designer framework that interfaces a development environment (Fig. 3, Programming environment 360), the hosting component and designer framework interface such that the unmanaged application functions as a designer within the development environment (Col. 6, lines 29-42 where application functions lists are located in tool bar buttons 363, main menu 362 and), the hosted application operates in at least two modes including a design mode and a runtime mode to distinguish between design actions and runtime actions within the hosted application. (Col. 5, lines 38-56 where development system 200, compiler 220 works as framework environment and Fig. 3, Object inspector window 391 design mode and code editor 381, Col. 6 lines 59-67, Col. 7 lines 27-35).

Regarding to claim 16, Herrmann teaches *a designer framework that facilitates interfacing the hosting component to the development environment* (see FIG 2. 211,201,202. col. 5, lines 36-55 where as the designer framework part of IDE component from source code and forms and the host component compiler is integrated is the part of designer framework.)

Regarding to claim 17 Herrmann teaches native functionality of the unmanaged application is available within the IDE (col.6, lines 43-58, where form 371 components are the basic building blocks of applications developed within the IDE)

Regarding to claim 18 Herrmann *facilitates adding a control to the document and editing properties of the control* (col. 6, lines 59-67 where 'properties' defines a component's appearance and window 391 to edit the form during design time)

Regarding to claim 19, Herrmann further teaches *facilitates a merging of menus of the unmanaged application and the IDE* (col. 6, 29-44 where as (FIG 3, 360) application development interface is part of developmental environment.)

Regarding to claim 20, Herrmann further teaches *the document contains at least one of embedded and linked objects* (col. 5, lines 36-55 where compiler works as a embedded and OLE linked object to use an application interface within IDE development tools.)

Regarding to claim 21, Herrmann teaches *the presentation of a properties window that displays properties for code-behind project items* (col. 6, lines 42-68 where as the form is a properties of a window item, which is part of the user interface within IDE.

Regarding to claim 22, Herrmann teaches *the designer operates in a design view and a code view* (col. 1, lines 36-50, col. 2, lines 42-50 where as source code is edited in design view.)

Regarding to claim 23, Herrmann teaches *the designer can handle the document in at least one of a binary format and an XML format* (Col. 11, lines 10-35 where as designer application HTML views the document in binary format. XML is the extended version of HTML, XML is a text-based designer within the intergraded development tools of HTML.)

Regarding claim 24, Herrmann discloses a computer-readable medium having computer-executable instructions for performing a method of integrating an unmanaged

Art Unit: 2192

application into a development environment (Col. 4, lines 59 to col. 5, lines 25), the method comprising:

Receiving the unmanaged application (col.3, lines 21-30); and interfacing the unmanaged application to the development environment (Fig. 3, form 371, Col. 3, lines 21-30 where ActiveX and Delphi are such example of an unmanaged application) with a host adaptor that is specific to the unmanaged application such that the unmanaged application is accessible as a designer within the development environment (col. 5, lines 6-55, see FIG 1B. 150, see FIG 2. 211,220,210 where computer readable medium is system memory, host adaptor is the compiler and unmanaged application is word processor, all of these are within the IDE.) a host adaptor (Fig. 3, main window 361) that merges at least one menu of the unmanaged application with at least one menu of the development environment (Col. 6, lines 33-42 where main menu 362 lists user-selectable command...), determines if the development environment has focus Or if unmanaged application has focus (col. 6, lines 43-47), and enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination (col. 6, lines 29-42, 51-58);

Regarding to claim 25, Herrmann teaches *hosting a document in the development environment such that the document is manipulated using native functionality of the unmanaged application and functionality of the development environment* (Fig. 3, application development environment 360, col. 6, lines 29-58 where main menu window 361 and its components such as: file, edit, view...).

Regarding to claim 26, Herrmann teaches *hosting a document in the development environment and exposing a code-behind file associated with the document such that contents of the file can be manipulated* (col. 6, lines 29-68 where as form is a content of host document as code editor within development environment.)

Regarding to claim 27, Herrmann teaches *hosting a document in the development environment and exposing a code-behind file associated with each subdocument of the document such that contents of each file can be manipulated* (col. 6, lines 29-68 where as subdocument are the main menu, file, edit, command and so on within the document toolbar for editing.)

Regarding to claim 28, Herrmann teaches *providing a view control that triggers in response to an event* (col. 11, lines 10-25, Col. 9, lines 14-31 where the user clicking on a hyperlink initiate a trigger for an event within the HTML).

Regarding claim 29, Herrmann discloses a method of integrating an unmanaged application into a development environment, comprising:

~~Receiving~~ obtaining *the unmanaged application* (Col 2. lines 35-55 where ActiveX is an unmanaged application); *interfacing the unmanaged application to the development environment with a host adaptor that is specific to the unmanaged application* (Fig. 3, form 371, Col. 3, lines 21-30 where ActiveX and Delphi are such example of an unmanaged application) *such that the unmanaged application is accessible as a designer within the development environment* (col. 5, lines 38-56 where compiler 220 as host adapter, interface 211, IDE 211...) a host adaptor that merges at

least one menu of the unmanaged application with at least one menu of the development environment (Fig. 3, Programming environment 360), determines if the development environment has focus OR if unmanaged application has focus (col. 6, lines 43-47), and enables at least one merged menu item (Col. 6, lines 33-42 where main menu 362 lists user-selectable command...) and disabled at least one merged menu item based upon a result of the determination (col. 6, lines 29-42, 51-58);

Hosting a document in the development environment such that functionality of the development environment (Col. 6, lines 29-42 where application functions lists are located in tool bar buttons 363, main menu 362 and) *and the unmanaged application can be used to manipulate the document* (col. 6, lines 29-68 where as form is a content of host document as code editor of an application can be used for editing within development environment.)

Regarding to claim 30, Herrmann teaches *comprising manipulating the document by at least one of, performing a drag-and-drop operation of a control onto the document* (col. 6, lines 50-67 where as application can change in design mode), *drag-and-drop can be perform within the design view; viewing a data-behind file associated with the document* (col. 2, lines 30-34); *writing managed code in the data-behind file* (col. 2, lines 51-65 where ActiveX and java applets runs in background); *operating the designer in a design mode and a runtime mode* (col. 5, lines 38-56); *presenting non-visual data and drawing managed objects on the document* (col. 6, lines 32-43).

Regarding to claim 31, Herrmann teaches *providing a special page of preferences for the unmanaged application such that when integrated into the*

development environment, the unmanaged application behaves according to the preferences (Col. 5, lines 36-56; where as preference page is a pre-defined libraries and link for the source code.)

Regarding claim 32, Herrmann teaches a system that facilitates of integrating an unmanaged application into a development environment, comprising:

Means for ~~receiving~~ obtaining the unmanaged application (Col 2. lines 35-55 where ActiveX is an unmanaged application, Fig 1A.100-107);

Means for interfacing the unmanaged application to the development environment (Fig. 3, Programming environment 360) with a host adaptor that is specific to the unmanaged application such that the unmanaged application is accessible as a designer within the development environment (col. 5, lines 6-55); a host adaptor that merges at least one menu of the unmanaged application (Col. 6, lines 33-42 where main menu 362 lists user-selectable command...) with at least one menu of the development environment, determines if the development environment has focus or if unmanaged application has focus (col. 6, lines 43-47), and enables at least one merged menu item and disabled at least one merged menu item based upon a result of the determination (col. 6, lines 29-42, 51-58);

Means for hosting a document in the development environment such that functionality of the development environment (Col. 6, lines 29-42 where application functions lists are located in tool bar buttons 363, main menu 362 and) and the unmanaged application can be used to manipulate the document; and means for manipulating the document (col. 5, lines 6-35, col. 6, lines 29-68).

Regarding to claim 33, Herrmann teaches *the means for manipulating further comprising manipulating means for at least one of* (col.1, lines 29-54, see FIG. 7), *performing a drag-and-drop operation of a control onto the document* (col. 6, lines 50-67); *viewing a data-behind file associated with the document* (col. 2, lines 30-34); *writing managed code in the data-behind file* (col. 2, lines 55-65); *operating the designer in a design mode and a runtime mode* (col. 5, lines 38-56); *presenting non-visual data and drawing managed objects on the document* (Col. 6, lines 32-43).

Regarding to claim 34, Herrmann teaches *means for exposing properties and components of the document for manipulation* (Col 6. lines 30-58).

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to AHMED E. HUQ whose telephone number is (571)270-1515. The examiner can normally be reached on Monday-Friday 9:-5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ahmed E Huq/
Examiner, Art Unit 2192
6/20/2008

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192